Amendments to the Claims:

Please replace the claims with the following listing of claims.

1. (Previously presented) A method for ensuring client access to unpaired messages from a database management system (DBMS) server, comprising:

   the DBMS server determining that a transaction response message is in response to a communication disruption between a client and the DBMS server, the DBMS server storing the unpaired message in an unpaired message queue, the unpaired message corresponding to a specific client, the transaction response message associated with a transaction request;

   creating the unpaired message queue in a DBMS server separate from a paired message queue in response to determining a transaction response message is an unpaired message, the unpaired message queue configured to store a plurality of unpaired messages intended for the client, the unpaired message queue and paired message queue storing messages according to a First-In-First-Out protocol; and

   communicating transaction requests and transaction responses using an Open Transaction Manager Access (OTMA) protocol which allows the client to request at least one unpaired message stored in the unpaired message queue.

2. (Previously presented) The method of claim 1 further comprising the DBMS server dynamically creating the unpaired message queue in response to the DBMS server detecting at least one unpaired message.

3. (Previously presented) The method of claim 1, further comprising notifying the DBMS server of a client request to enable dynamic creation of the unpaired message queue.

4. (Previously presented) The method of claim 3, wherein notifying the DBMS server occurs during establishment of communications between the client and the DBMS server.

5. (Previously presented) The method of claim 1, further comprising the DBMS server notifying the client when the unpaired message queue contains an unpaired message.

6. (Previously presented) The method of claim 1, further comprising:

   generating a request message to be sent from the client to the DBMS server; and storing an indicator in the request message to enable the client to distinguish between unpaired messages.

7. (Previously Presented) The method of claim 1, wherein utilizing the protocol further comprises allowing the client to request automatic transmission of unpaired messages stored in the unpaired message queue.

8. (Currently amended) A computer program product comprising a tangible computer ~~readable~~ useable medium having stored thereon computer executable instructions for performing a method for ensuring client access to unpaired messages from a database management system (DBMS) server, the computer useable medium operable within a memory device and executable on a processor, the computer program product ~~method~~ comprising:

the DBMS server determining that a transaction response message is an unpaired message in response to a communication disruption between a client and the DBMS server, the DBMS server storing the unpaired message in an unpaired message queue, the unpaired message corresponding to a specific client, the transaction response message associated with a transaction request;

creating the unpaired message queue in ~~a~~ the DBMS server separate from a paired message queue in response to determining a transaction response message is an unpaired message, the unpaired message queue configured to store a plurality of unpaired messages intended for a client, the unpaired message queue and paired message queue storing messages according to a First-In-First-Out protocol; and

communicating transaction requests and transaction responses using an Open Transaction Manager Access (OTMA) protocol which allows the client to request at least one unpaired message stored in the unpaired message queue.

9. (Currently amended) The computer program product~~readable medium~~ of claim 8, wherein the method further comprising the DBMS server dynamically creating the unpaired message queue in response to the DBMS server detecting at least one unpaired message.

10. (Currently amended) The computer program product~~readable medium~~ of claim 8, wherein the method further comprises notifying the DBMS server of a client request to enable dynamic creation of the unpaired message queue.

11. (Currently amended) The computer program product~~readable medium~~ of claim 10, wherein notifying the DBMS server occurs during establishment of communications between the client and the DBMS sever.

12. (Currently amended) The computer program product~~readable medium~~ claim 8, wherein the method further comprises the DBMS server notifying the client when the unpaired message queue contains an unpaired message.

13. (Currently amended) The computer program product~~readable medium~~ of claim 8, wherein the method further comprises:

    generating a request message to be sent from the client to the DBMS server;

    storing an indicator in the request message to enable the client to distinguish between unpaired messages.

14. (Currently amended) The computer program product~~readable medium~~ of claim 8, wherein utilizing the protocol further comprises allowing the client to request automatic transmission of unpaired messages stored in the unpaired message queue.

15. (Currently amended) A system for ensuring client access to unpaired messages from a database management system (DBMS) server comprising:

    a DBMS server comprising a memory and a processor;

    a client;

    a request module configured to receive a client request;

a response generator which receives the client request from the request module and generates an appropriate transaction response message generated based on the client request, the transaction response message associated with a transaction request;

an unpaired message module that determines that the transaction response message is an unpaired message in response to a communication disruption between a client and the DBMS server and to store paired messages in a paired response data structure and unpaired messages in an unpaired response data structure in response to determining a transaction response message is an unpaired message, the at least one unpaired message comprising a communication response for a specific client, the unpaired response data structure and paired response data structure storing messages according to a First-In-First-Out protocol; and

a response module which communicates paired messages and unpaired messages to a client configure with simple communication logic that conforms to ~~a~~an Open Transaction Manager Access (OTMA) protocol such that the client constitutes a thin client.

16. (Original) The system of claim 15, wherein the unpaired message module is further configured to dynamically create the unpaired response data structure in response to a first unpaired response message.
17. (Original) The system of claim 15, wherein the response module is configured to automatically send all unpaired messages stored in the unpaired response data structure.
18. (Original) The system of claim 15, wherein the response module is configured to send all unpaired messages stored in the unpaired response data structure in response to a request from the client.
19. (Previously presented) The system of claim 15, wherein the system is activated upon the DBMS server receiving an activation request from the client.
20. (Original) The system of claim 15, wherein the response module notifies the client when the unpaired response data structure contains at least one unpaired message.